

LAPORAN JOBSHEET 12

Dasar Pemrograman Searching dan Sorting

Dibina oleh Ibu Vivi Nur Wijyaningrum, S.Kom., M.Kom.



Disusun oleh :

Fitria Ramadhani Prihandiva

D-IV Sistem Informasi Bisnis 1C / 14

2241760055

POLITEKNIK NEGERI MALANG

JURUSAN TEKNOLOGI INFORMASI

TAHUN 2023

Nama : Fitria Ramadhani Prihandiva
Kelas : SIB 1C
NIM : 2241760055
Mata Kuliah : Praktikum Dasar Pemrograman

JOBSHEET 12

Percobaan 1 : Unordered Sequential Search pada Array

```
public class UnorderedSearch {  
    static int search (int[]arr, int num){  
        for (int i = 0; i<arr.length; i++){  
            if (arr[i] == num)  
                return i;  
        }return -1;  
    }  
}  
Run | Debug  
public static void main (String []args){  
    int[] arrInt = {10,5,20,2,11,8,4,15};  
    int cari = 8;  
    int hasil = search(arrInt,cari);  
    if (hasil == -1)  
        System.out.println("Elemen " + cari + " Tidak ditemukan");  
    else  
        System.out.println("Elemen " + cari + " ditemukan pada posisi ke-" + hasil);  
}
```

```
D:\College\SEMESTER 2\Daspro\Jobsheet 12>java UnorderedSearch.java  
Elemen 8 ditemukan pada posisi ke-5
```

Pertanyaan

1. Pada percobaan di atas, jika cari = 8 dan cari = 7, maka masing-masing berapa kali banyak perbandingan yang dilakukan?

Jawaban : Karena data 7 tidak ada maka pencarian dilakukan sebanyak 8x berurutan dari depan hingga kebelakang untuk mencari angka 7.

2. Modifikasi kode program di atas dengan menerima nilai variabel **cari** secara dinamis (berdasarkan nilai yang dimasukkan pengguna)!

Jawaban :

```

import java.util.Scanner;
public class UnorderedSearch {
    static int search (int[]arr, int num){
        for (int i = 0; i<arr.length; i++){
            if (arr[i] == num)
                return i;
        }return -1;
    }
}

Run | Debug
public static void main (String []args){
    Scanner fit = new Scanner(System.in);
    int[] arrInt = {10,5,20,2,11,8,4,15};
    System.out.println("Masukkan angka yang anda cari: ");
    int cari = fit.nextInt();
    int hasil = search(arrInt,cari);
    if (hasil == -1)
        System.out.println("Elemen " + cari + " Tidak ditemukan");
    else
        System.out.println("Elemen " + cari + " ditemukan pada posisi ke-" + hasil);
}

```

```

D:\College (SEMESTER 2)\Daspro\001
Masukkan angka yang anda cari:
40
Elemen 40 Tidak ditemukan

```

Saya menambahkan **scanner** yang disimpan di variable **cari** sehingga pengguna bisa memasukkan angka yang ingin dicari kemudian angka tersebut akan tersimpan di variable cari untuk diproses melakukan pencarian.

Percobaan 2 : Unordered Sequential Search pada Array of Object

```

public class Mahasiswa {
    public String nim;
    public String namaMahasiswa;
    public double ipk;

    public Mahasiswa (String id, String name, double gpa){
        nim = id;
        namaMahasiswa= name;
        ipk = gpa;
    }
}

```

```

import java.util.Scanner;

import Java.util.scanner;
public class MahasiswaMain {
    Run | Debug
    public static void main (String[]args){
        Mahasiswa [] mhs = new Mahasiswa[10];
        mhs [0] = new Mahasiswa (id:"19970506001", name:"Alfatih", gpa:3.9);
        mhs [1] = new Mahasiswa (id:"19910421002", name:"Gitta", gpa:3.8);
        mhs [2] = new Mahasiswa (id:"19950322002", name:"Amanda", gpa:3.75);
        mhs [3] = new Mahasiswa (id:"19980129001", name:"Kevin", gpa:3.5);
        mhs [4] = new Mahasiswa (id:"19990208002", name:"Windy", gpa:3.92);
        mhs [5] = new Mahasiswa (id:"19970710001", name:"Belva", gpa:3.6);
        mhs [6] = new Mahasiswa (id:"19920602001", name:"James", gpa:3.7);
        mhs [7] = new Mahasiswa (id:"19901106002", name:"Aruna", gpa:3.3);
        mhs [8] = new Mahasiswa (id:"19981005002", name:"Sabrina", gpa:3.55);
        mhs [9] = new Mahasiswa (id:"19991201001", name:"Agus", gpa:3.65);

        Scanner fit = new Scanner (System.in);
        System.out.println(x:"Masukkan nama mahasiswa yang dicari: ");
        String cari = fit.nextLine();
        for (int i=0; i<mhs.length; i++){
            if (cari.equalsIgnoreCase(mhs[i].namaMahasiswa)){
                System.out.println("Nama " + mhs[i].namaMahasiswa);
                System.out.println("NIM " + mhs[i].nim);
                System.out.println("IPK " + mhs[i].ipk);
            }
        }
    }
}

```

```

Masukkan nama mahasiswa yang dicari:
Agus
Nama Agus
NIM 19991201001
IPK 3.65
PS C:\Users\FITRIA RAMADHANI>

```

```

Masukkan nama mahasiswa yang dicari:
gitta
Nama Gitta
NIM 19910421002
IPK 3.8

```

Pertanyaan

1. Perhatikan class **MahasiswaMain**, jelaskan fungsi kode program berikut ini!

```
if (cari.equalsIgnoreCase(mhs[i].namaMahasiswa))
```

Jawaban : Untuk melakukan pengecekan dengan kondisi nama yang dicari/input sama dengan yang sudah diinisialisasi. Nama yang diinputkan disimpan pada variable **cari** kemudian dilakukan pengecekan dengan **equals.IgnoreCase** dengan tujuan untuk mengabaikan besar kecil huruf inputannya. Pencarian dilakukan di Array **Mahasiswa** dengan nama objek **mhs** pada atribut **namaMahasiswa**.

2. Lakukan modifikasi pada program tersebut, sehingga jika data yang dicari tidak ditemukan, maka akan menampilkan "Data tidak ditemukan"!

Jawaban :

```

Scanner fit = new Scanner (System.in);
System.out.println(x:"Masukkan nama mahasiswa yang dicari: ");
String cari = fit.nextLine();
boolean no = false;
for (int i=0; i<mhs.length; i++){
    if (cari.equalsIgnoreCase(mhs[i].namaMahasiswa)){
        System.out.println("Nama " + mhs[i].namaMahasiswa);
        System.out.println("NIM " + mhs[i].nim);
        System.out.println("IPK " + mhs[i].ipk);
        no=true;
    }
}
if(!no) {
    System.out.println(x:"Data tidak ditemukan");
}

```

```

Masukkan nama mahasiswa yang dicari:
fitria
Data tidak ditemukan

```

Menambahkan **System.out.println** diluar outer loop. Apabila diletakkan di dalam perulangan maka akan menghasilkan seperti di bawah ini. Sistem akan terus melakukan print setiap pencarian data tidak ditemukan. Kemudian menambah variable bertipe boolean dengan inisialisasi false. Variable tersebut dipanggil lagi ke If yang dengan kondisi !no agar bisa mengecek data yang masuk apabila nama salah maka akan muncul data tidak ditemukan

```

Masukkan nama mahasiswa yang dicari:
fitria
Data tidak ditemukan
Data tidak ditemukan
Data tidak ditemukan
Data tidak ditemukan
Data tidak ditemukan
Data tidak ditemukan
Data tidak ditemukan
Data tidak ditemukan
Data tidak ditemukan
Data tidak ditemukan

```

3. Lakukan modifikasi pada program tersebut agar pencarian dilakukan bukan berdasarkan nama mahasiswa, melainkan berdasarkan IPK!

```

System.out.println("Masukkan nilai IPK yang anda cari: ");
double cari2 = fit.nextDouble();
fit.nextLine();
for (int j=0; j<mhs.length; j++){
    if (cari2 == mhs[j].ipk){
        System.out.println("Nama " + mhs[j].namaMahasiswa);
        System.out.println("NIM " + mhs[j].nim);
        System.out.println("IPK " + mhs[j].ipk);
    }
}

```

```

Masukkan nilai IPK yang anda cari:
3.75
Nama Amanda
NIM 19950322002
IPK 3.75

```

Sama seperti yang mencari nama, namun memperhatikan tipe datanya, yang awalnya String dirubah menjadi double menyesuaikan tipe data dari IPK.

Percobaan 3 : Ordered Sequential Search pada Array of Object

```

public class OrderedSearch {
    public double[] arr;

    public OrderedSearch (double arrayNilai[]){
        arr = new double[arrayNilai.length];
        for(int i = 0; i<arr.length; i++){
            arr[i]=arrayNilai[i];
        }
    }

    public int cari (double keyword){
        int index = -1;
        for(int i =0 ; i<arr.length; i++){
            if (keyword == arr[i]){
                index = i;
                break;
            }else{
                if(keyword<arr[i]){
                    break;
                }
            }
        }
        return index;
    }

    public void tampilkan(){
        for(int i=0; i<arr.length;i++){
            System.out.print(arr[i] + " ");
        }
        System.out.println("\n");
    }
}

```

```

public class PencarianMain {
    Run | Debug
    public static void main (String[] args){
        double[] data = {2,7,2.9,3.04,3.12,3.54,3.87,3.9};
        OrderedSearch os = new OrderedSearch(data);
        System.out.println("Isi elemen array: ");
        os.tampilkan();
        double key=3.12;
        int index = os.cari(key);
        if (index != -1){
            System.out.println("Data " + key + " pada index " + index);
        }else{
            System.out.println("Data " + key+ " tidak ditemukan");
        }
    }
}

```

```
Isi elemen array:  
2.0 7.0 2.9 3.04 3.12 3.54 3.87 3.9  
Data 3.12 tidak ditemukan
```

Pertanyaan

1. Perhatikan class **PencarianMain**, jelaskan fungsi dari kode kode program berikut ini!

```
OrderedSearch os= new OrderedSearch(data);
```

Jawaban : Inisialisasi object os yang isinya data

2. Perhatikan class **Pencarian**, pada method **Cari** terdapat dua kali **break**. Jelaskan fungsi masingmasing break tersebut pada kode program tersebut!

Jawaban : Break pertama untuk menghentikan looping ketika nilai yang dicari sudah ditemukan dari nilai pada indeks ke-i. Sedangkan break kedua digunakan untuk menghentikan proses perulangan ketika nilai yang dicari lebih kecil pada indeks ke-i.

Percobaan 4 : Sorting menggunakan Bubble Sort


```

public class Sort {
    public int [] data;
    public int jumData;

    public Sort(int Data[], int jmlData){
        jumData = jmlData;
        data= new int [jmlData];
        for (int i =0; i<jumData; i++){
            data[i]=Data[i];
        }
    }

    public void bubbleSort(){
        int temp=0;
        for(int i=0; i<(jumData-1);i++){
            for(int j=1; j<(jumData-i); j++){
                if(data[j-1]>data[j]){
                    temp=data[j];
                    data[j]=data[j-1];
                    data[j-1]=temp;
                }
            }
        }
        tampilData();
    }

    public void tampilData(){
        for(int i=0; i<jumData; i++){
            System.out.println(data[i]+ " ");
        }
        System.out.println();
    }

    Run | Debug
    public static void main(String[]args){
        int a[] = {15,10,7,22,5};
        Sort urut = new Sort(a,a.length);
        System.out.println("Data Sebelum di urutkan");
        urut.tampilData();
        System.out.println("Data Setelah di urutkan");
        urut.bubbleSort();
    }
}

```

Data Sebelum di urutkan

15
10
7
22
5

Data Setelah di urutkan

5
7
10
15
22

Pertanyaan

1. Tunjukkan pada bagian mana proses pertukaran data terjadi dalam Bubble Sort!

```
if(data[j-1]>data[j]){  
    temp=data[j];  
    data[j]=data[j-1];  
    data[j-1]=temp;  
}
```

Jawaban :

2. Jelaskan fungsi penggunaan nested loop pada method **bubbleSort()**!

Jawaban : Innerloop melakukan perulangan untuk menukar elemen elemen secara bergantian. Sedangkan outerloop melakukan perulangan yang dari awal lagi (seperti melakukan pengecekan lagi) jika ditemukan ada yang masih tidak urut, maka masuk lagi ke innerloop untuk melakukan penukaran elemen. Hingga elemen telah terurut